SUMMARY OF RESEARCH

Modifying and Testing ATC Controller Interface (CI) for Data Link Clearances

Grant: NAG-1-2036

by

Avionics Engineering Center
School of Electrical Engineering and Computer Science
Ohio University
Athens, Ohio 45701-2979

September 2001

Prepared for:

NASA-Langley Research Center
Hampton VA 23681-2199

Table of Contents

Modifying and Testing ATC Controller Interface (CI) for Data Link Clearances

1 Overview

NASA Grant NAG-1-2036 titled "Modifying and Testing ATC Controller Interface (CI) for Data Link Clearances" enhanced the Controller-Pilot Datalink Communications (CPDLC) and Air Traffic Control workstation research that was conducted as part of the 1997 NASA Low Visibility Landing and Surface Operations (LVLASO) demonstration program at Atlanta Hartsfield airport.

Research activity under this grant increased the sophistication of the Controllers' Communication and Situational Awareness Terminal (C-CAST) and developed a VHF Data Link –Mode 2 communications platform. The research culminated with participation in the 2000 NASA Aviation Safety Program's Synthetic Vision System (SVS) / Runway Incursion Prevention System (RIPS) flight demonstration at Dallas-Fort Worth Airport.

2 Organization

The C-CAST team was led by Dr. James Rankin, Principal Investigator, Ohio University Avionics Engineering Center (AEC). Sanjeev Gunawardena, AEC Research Engineer, led the development of the VDLM2 datalink. Ohio University students, Eric Best and Qingwei Ma, assisted with the C-CAST software development.

Pat Mattson, Department of Aviation, St. Cloud State University, St. Cloud MN, was the technical lead for the C-CAST voice recognition system. Mr. Mattson also provided knowledge of the Air Traffic Controller's work environment. St. Cloud State students, Alicia Lechner and Kevin Ecker, assisted Mr. Mattson with the software development.

3 Background

With the consistent traffic increase in today's skies, the current system is being tested to its limits. A large amount of research is being conducted to help combat the problem of increasing complexity of today's ATC systems. One of the major areas of research is ATC automation. This will help take some of the burden from ATC controllers and pilots allowing them to concentrate solely on the issues that require human intervention. C-CAST is one possible tool for this operation.

3.1 Controller/Pilot Communications

The most common ATC message is to contact some other ATC facility on a particular frequency. This type of repetition can take up a lot of time, making the system inefficient from a time standpoint. A solution to this problem is the use of digital datalink communications and voice recognition to create and send the message quickly. This is considerably faster than having to spell out the entire message verbally and reduces the probability of a miscommunication.

By implementing this type of system, the controller would be able to handle more aircraft in a fraction of time with less effort than before. They would not have to speak messages hundreds of times a day, reducing the effect of fatigue from constant repetition. This would make the controller more effective and possibly more alert.

3.2 ATC Efficiency

The ability to speed-up communication can allow a faster aircraft turnover rate. This faster communication does not include the benefits provided by the other entities of the C-CAST system. C-CAST not only provides the controller with a faster means of communication, but it also has an easy to read visual interface and provides safety benefits as well.

3.3   Radio Frequency Congestion

Datalink communications technology is being pushed as a partial remedy for radio frequency congestion. Routine instructions and acknowledgements communicated via datalink take a fraction of the time required by current voice message systems. This can range from seconds for a data-link to even minutes in some voice environments. Time can be saved in two ways using a data-link. First, the digital messages are encoded so that relatively large messages are transmitted using only a few bits of data. Second, all spoken communication that is long-winded or cryptic, heavily accented, stepped upon, repetitious and often ambiguous or misunderstood are removed.

4   Research

Research funded by this grant led to the modification and enhancement of the Controller Interface (CI) used at Atlanta. The new C-CAST system was improved through the use of new 32-bit oriented software design, TCP/IP interfaces instead of slower RS-232 interfaces, and more robust software that could handle additional research features. New features in the C-CAST included the ability to uplink runway environment messages to the Hold Short and Landing Technology (H-SALT) system, the display of runway incursion alerts and warnings to the controller, and the inclusion of "fused" traffic data from the surveillance server. The C-CAST system is described in Section 5.

Another major research accomplishment was the design, development, and test of a VHF Datalink – Mode 2 (VDLM2) communications channel. The development of Controller-Pilot Data Link Communications (CPDLC) in the U.S. and abroad are initially implementing the system with a VDLM2 datalink. Research conducted at Ohio University led to the successful use of a VLDM2 for CPDLC messages on the airport surface. The VDLM2 communications channel included the development of a Datalink Manager as part of the surface infrastructure and an airborne datalink manager on the NASA B757. The VDLM2 system is described in Section 6.

The third research area was a voice recognition system. The 1997 Atlanta system used a voice recognition card by Verbex. This card provided excellent results but had several drawbacks. The card would not work with 32-bit software and the grammar file required each controller to train the system.   At DFW, a Lernout & Hauspie (L&H) system was used. The L&H software did not require a special computer card, but operated on the PC sound card that was already installed. The L&H voice recognition system was also speaker-independent, which means that a user does not have to train the system. The voice recognition system is described in Section 7.

5   Controllers' Communication and Situational Awareness Terminal (C-CAST)

The C-CAST system is a ground or local controller's workstation that provides electronic flight strips, CPDLC user interface, aircraft and ground vehicle position display, and runway incursion alerting. A snapshot of C-CAST can be seen in Figure 1. C-CAST receives traffic information from a Surveillance Server via a wireless LAN from the DFW East Control Tower. A C-CAST Data Link Manager (DLM) connects all ground systems and routes all the ground data. All ground systems are connected to a Network Time Protocol (NTP), which time stamps all data transfers for synchronization and error detection purposes. For the 2000 DFW flight test, the C-CAST system was located in a pseudo-ATC control room located in the Harvey Hotel.

All these systems combine to form the NASA Runway Incursion Prevention System. C-CAST is used on the ground, by a controller, to see the traffic on the runways and in final approach and allows them to send commands to any aircraft that has established a communication link. A controller can send a message to the aircraft via the voice recognition system or via an easy to use manual interface. Manual entries can be entered using a touch screen monitor or a mouse if preferred. After the message is sent to the aircraft, it is then translated and displayed for the pilot. This serves as a visual backup in case of a misheard or misunderstood command.
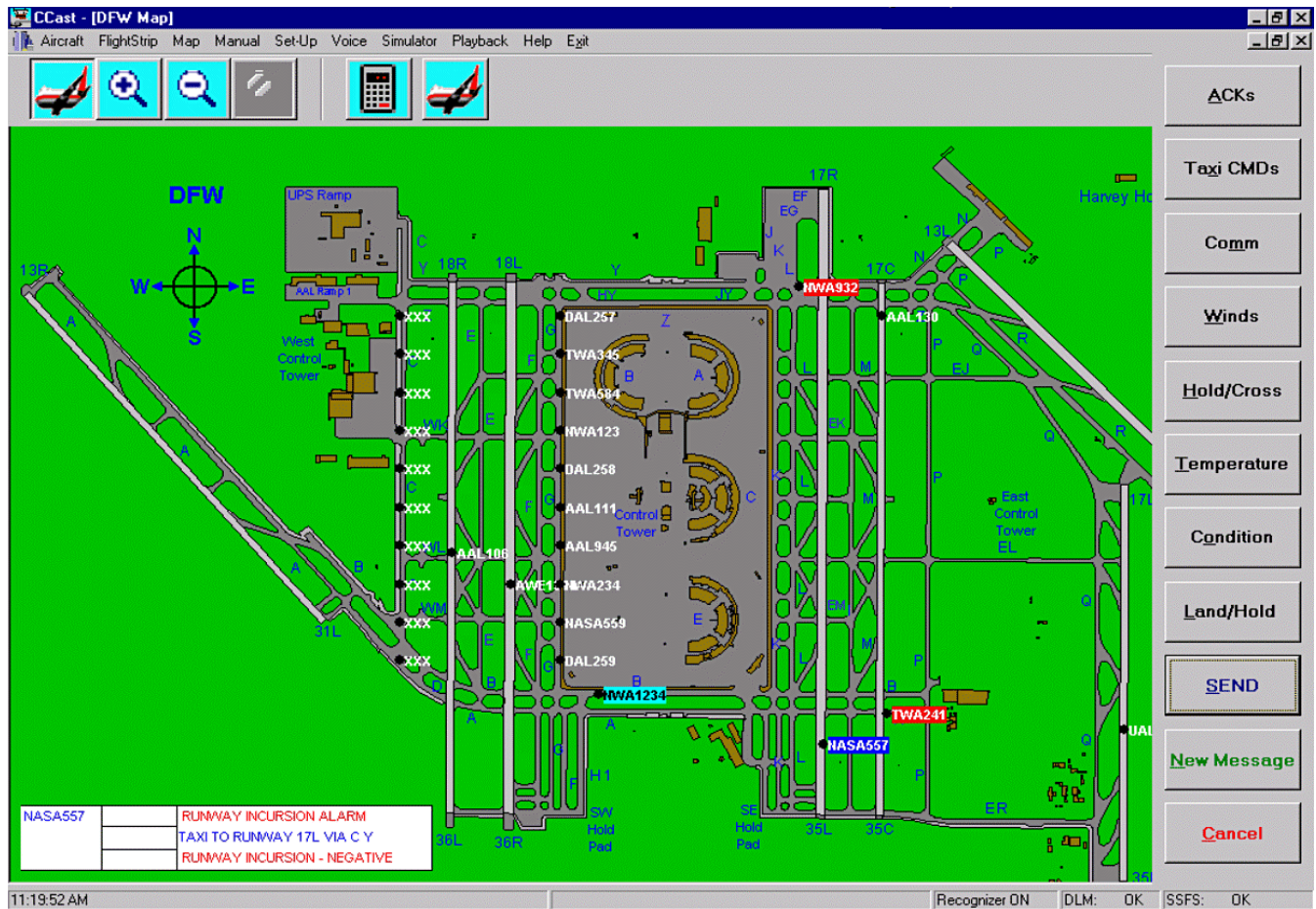
Figure 1 . C-CAST display

5.1   Manual Interface

The C-CAST manual interface serves as a reversionary mode in case the voice recognition system becomes inoperative. The Manual interface allows the user to build and send any possible message to the aircraft through a series of menus. There is a strip of buttons on the right side of the screen that can be manipulated via the touch screen or with a mouse. These buttons are the top-level menus for all the air traffic messages needed by the controller. A user must first select an active aircraft by highlighting its call sign and is then allowed to build a message.

Each top-level button has a series of sub-buttons that make up the entire message. Depending on the message, there may be several data entries needed in order to build the entire command. Once a message is created, it is displayed in the center of the status bar in blue text. At this stage the message can either be sent or cleared for another entry.

5.2   Zoomable Map

The map feature in C-CAST allows the user to see all aircraft that are detected by the local surveillance system. The traffic data is supplied to C-CAST from the Surveillance Server and is updated twice a second. A full map view of the entire airport is available as well as zoomed views of anywhere on the map. A simple press of a button can restore the map to its original state.

The map used on C-CAST was provided by Jeppesen-Sanderson via NASA-LaRC. The map was created from an accurate (1' resolution) survey of the airport area.

A valuable asset supplied with C-CAST is the presence of hold bars. Red or yellow lines appear at the intersections of runways and taxiways that indicate vehicles are not supposed to cross due to other aircraft approaching the intersection in question. Both the controller and pilot are able to see these lines on their displays. This will help reduce the frequency of runway incursions, making the runways safer and more efficient.

5.3   Binary Message Format

All messages entered on the C-CAST, either by voice or manually, are translated into the ICAO ATN formats. These are standardized formats for ATC Two-Way Data-Link Communications. Each command has its own code number, called its "Element ID". This message is then imbedded into a message packet used by the DLM to transmit the message.

Table 1 shows the uplink message Element Ids and Table 2 has the Downlink ElementIDs. The Element ID is the first data entered into the message after the header. After the Element ID, the rest of the message is appended as a function of the information necessary to complete the respective message. This follows the ATN-SARPS format as obtained from the 1999 2nd edition.

Table 1   Uplink Elment IDs

Element ID	Meaning
0	UNABLE
1	STANDBY
2	REQUEST DEFERRED
3	ROGER
117	CONTACT [icaounitname] [frequency]
120	MONITOR [icaounitname] [frequency]
153	ALTIMETER [altimeter]
169	[freetext]
240	HOLD SHORT OF [position]
241	TAXI RUNWAY [runway] VIA [taxiroute]
242	TAXI RAMP [ramp] VIA [taxiroute]
243	CROSS [position] [without delay]
244	CONTINUE TAXI
245	UNAVAILABLE TAXIWAYS [taxiways]
246	RUNWAY [runway] TAXI INTO POSITION AND HOLD
247	RUNWAY [runway] CLEARED FOR TAKEOFF
248	WIND [direction] AT [speed]
249	RUNWAY CONDITION [condition]
250	LAND AND HOLD SHORT OF [runway]
251	TAXI TO GATE [gatenumber] VIA [taxiroute]
252	TEMPERATURE [temperaturec]
253	RUNWAY [runway] CLEARED TO LAND
254	TAXI TO SPOT [spotnumber] VIA [taxiroute]

Table 2 Downlink Messages

Element ID	Message
1	UNABLE
3	ROGER
102	LANDING REPORT
201	REQUEST TAXI CLEARANCE
202	TAXI DEVIATION
203	TURNED-OFF ON TAXIWAY
204	TAXI DEVIATION RESOLVED
205	RUNWAY INCURSION [source] [alarm type] [identification]
206	ASSIGNED GATE [gatenumber]

6   Datalinks

The wireless datalink used to send and receive CPDLC messages was a crucial part of the RIPS ground infrastructure. Figure 2 CPDLC Sub-system Architecture shows a high level diagram of the RIPS CPDLC subsystem. The C-CAST was physically located in the temporary NASA RIPS operational base located on the 15th floor of the Harvey Hotel, which had a clear view of the East side runways where the runway incursion test scenarios were performed. The C-CAST was connected to the CPDLC Datalink Manager (CDLM) located at the East Control Tower via a wireless Ethernet connection. The CDLM and its airborne counterpart, the Airborne Datalink Manager (ADLM) handled interfacing to the radio equipment used to implement the datalink. As shown in Figure 2, the ADLM communicated with the airborne unit's Onyx mainframe computer via the standard TCP/IP networking protocol.

The CPDLC datalink for the DFW tests utilized the new aeronautical VHF digital data communications system defined by RTCA Special Committee 172 (SC-172). Specifically, the datalink used VDL Mode 2 defined in the standard. Mode-S was used for CPDLC communications at Atlanta.
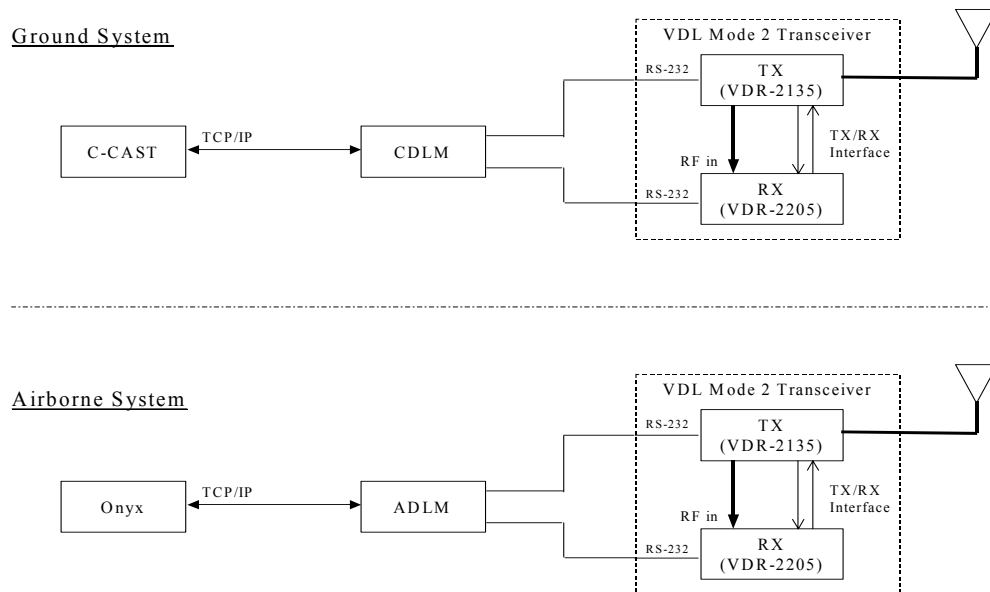

Figure 2 CPDLC Sub-system Architecture

6.1   VDL Mode 2 Standard and Equipment

As a replacement to the present double-sideband amplitude modulation (DSB-AM) based aeronautical VHF communications system operating within the radio frequency range 118.0 – 137.0 MHz, SC-172 describes the Minimum Aviation System Performance Standards (MASPS) for an advanced VHF digital datalink communications system including capability for digital voice techniques. Under this specification, two modes of operation are defined: VDL Mode 2 and VDL Mode 3.

VDL Mode 2 refers to the operation of a Carrier Sense Multiple Access (CSMA) scheme that supports datalink capabilities. CSMA allows multiple transmitters to share a single RF channel by having a connected receiver monitor the frequency for other transmissions. Once the channel is clear, a transmission is attempted based on a pre-configured set of statistical parameters. This statistical sharing of a channel results in efficient channel utilization. VDL Mode 2 is appropriate for aperiodic traffic where the entire message is ready before transmission of individual message packets is attempted. Due to the nature of CSMA, collisions are possible and hence this mode is not suited for time critical applications such as real time digital voice.

The radio equipment used to implement the datalink consisted of a Harris VDR-2205 receiver and Harris VDR-2135 transmitter pair configured as a single transceiver. Both the air and ground stations used identical equipment except that the airborne units were flight-hardened by NASA-Langley. The VDR-2135 transmitters feature an internal solid-state transmit/receive antenna switch that greatly simplified the configuration of the units as transceivers.

Interfacing to the radios was performed via their RS-232 data I/O port at 57.6 kbps. The datalink manager software supplied the Destination Address, Source Address, Control Byte, and the application (CPDLC) data components of an Aviation VHF Link Control (AVLC) frame. The transmitter completed the frame by inserting the Start-of-Frame flag, the End-of-Frame Flag, the Frame Check Sequence, the Mode 2 Training Sequence, and the Forward Error Correction (FEC) field to complete the AVLC packet before transmission. Even though the transmitter has the capability to transmit several such AVLC frames in a single RF burst, only single-frame bursts were used for the RIPS tests.

Upon receipt of an RF burst, the receiver checks the FEC coding embedded in the message and corrects errors that may have occurred in the RF transmission. The receiver also performs the de-interleaving, de-scrambling, and removal of the header and training sequences before transferring the message to its host buffer, from where it will be passed to the datalink manager. Following the handoff of the AVLC frame to the host, the receiver reports the following parameters pertaining to the received RF burst: signal quality, received signal strength (RSS), number of symbols received, number of bytes and blocks corrected with FEC algorithm, confidence factor, end of burst indication and a broken message indication. These parameters were logged by the datalink manager for subsequent datalink performance analysis.

6.2   VDL Mode 2 Coverage Testing at DFW

To ensure that the VDL Mode 2 datalink will work reliably for the RIPS CPDLC operations, coverage tests were performed at DFW during August 2000. A temporary ground station was setup and a van was used to simulate an aircraft on the surface of the airport. The airborne transceiver was installed in the van. The van antenna was a standard ¼ wave whip antenna placed on its roof using a magnetic mount. The datalink manager software that controlled the airborne transceiver was configured to transmit a CPDLC downlink "ROGER" message approximately every second. The ground station datalink manager was configured to reply to this message with the corresponding uplink "ROGER" message. The transmitted and received messages at both ends, along with their reported VDL Mode 2 parameters were time stamped and logged for analysis. The tests were performed on two consecutive nights when airport traffic was minimal. During the tests, the van traveled along most runways and taxiways on the East side of the airport, and some West side taxiways that were to be used for the RIPS flight tests. Airport surface coverage plots were generated by correlating the received messages' time stamps with GPS position and plotting that location's received signal strength. The coverage from ground-to-van (uplink) and van-to-ground (downlink) for the first day of testing are shown in Figure 3and Figure 4.

Results obtained were roughly the same for both days of testing. Comparison between the uplink versus downlink plots show that the downlink plots have slightly better received signal strength. The reason for this being that the ground antenna was a narrow-band type tuned exactly to the frequency of operation, hence providing higher gain. Careful observation of the plots reveals areas of severe signal attenuation due to obstructions. For example, the signal received at the mid-section of runway 31R and the corresponding Q and R taxiways is attenuated due to obstruction by the East control tower. Similar attenuation occurs on runway 35R due to obstruction by the Delta Airways hangar. In this respect, the terminal buildings on the center of the airport are the most significant obstructions since they block the signal path for much of the West side of the airport. Even though the received signal strength at the West side was frequently less than -90 dBm, no messages had bit errors and there were no missed messages detected for both days of testing. This proved the reliability and robustness of the VDL Mode 2 datalink and provided a high degree of confidence that the datalink would work well during the RIPS flight tests.
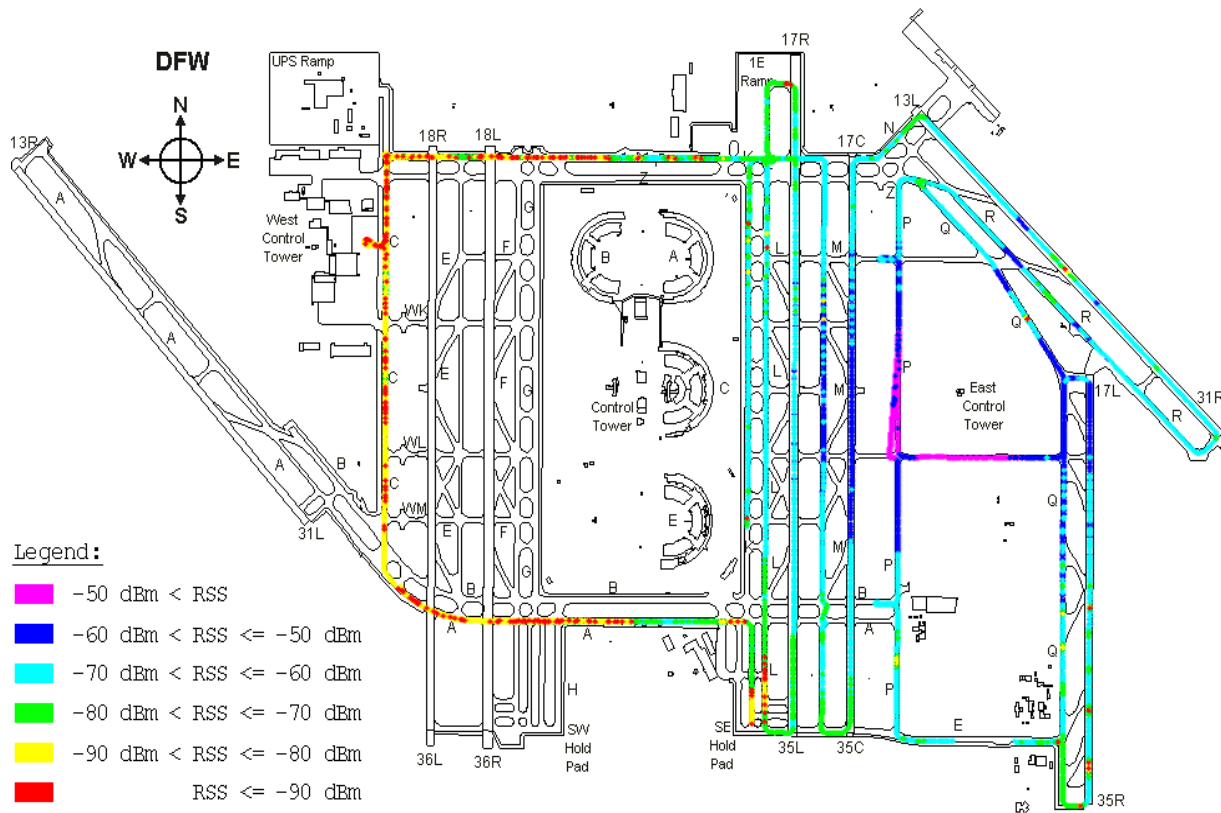

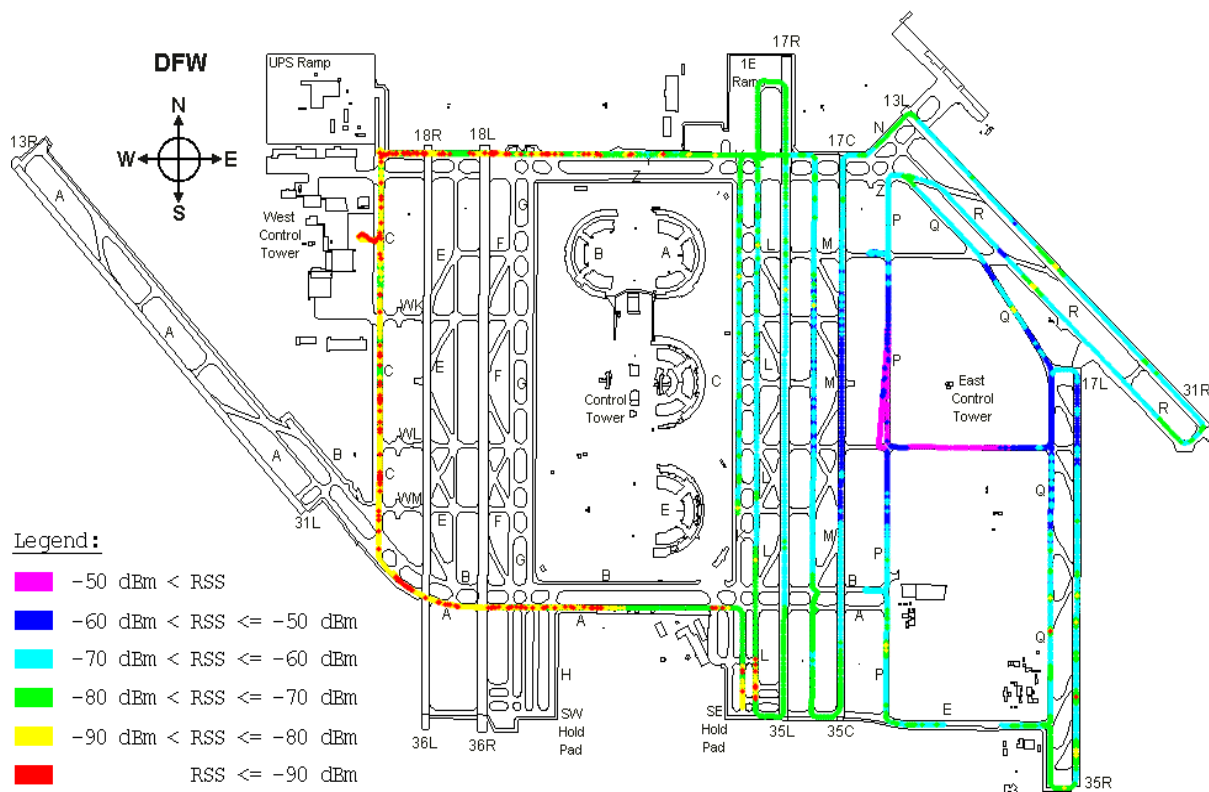**Figure 3 Ground-to-van (uplink) coverage 8-23-00**

Figure 4 Van-to-Ground (Downlink) Coverage 8-23-00

6.3 CPDLC Message Structure

Selected messages from the ICAO ATN SARPS plus new messages for the airport surface environment were implemented in the C-CAST. Following is a summary of these uplink (Table 1) and downlink messages (Table 2). These messages and their associated parameters were encoded into a variable length packet according to the encoding rules. The encoding rules are designed to minimize the overall length of the packet.

The CPDLC data packet was then embedded into a CPDLC message structure before being sent out to the CDLM or ADLM via TCP/IP. The format of this structure was largely influenced by the requirements of the VDL AVLC frame specification. Table 3 presents the CPDLC message structure.

Table 3 General CPDLC Message Structure

Data Field	Byte #
Message Type	0
Message Length (N)	1 : 2
Destination Address	3 : 6
Source Address	7 : 10
Control Byte (0x00)	11
Message Number (0 to 63)[*]	12
Aircraft ID (ASCII)[*]	13 : 20
CPDLC Packet (N Bytes)[*]	21 : N+2
C-CAST/Onyx Status[#]	12
Received Message Number[+]	12
Received Message Status[+]	13
Checksum (2 bytes)	Next 2 bytes
Integrity Byte (received messages only)	Last byte

The Message Type byte determined how the message was handled by the datalink nodes. In general, the Message Type range 0x10 to 0x1F was reserved for uplink messages (ground-to-air) and the range 0x20 to 0x2F was reserved for downlink (air-to-ground) messages. All message type definitions used for the RIPS tests are given in Table 4.

Since the CPDLC message structure had variable length, a Message Length field was needed so that the parsing software could properly identify the end of the message. The destination and source address fields contained the 23-bit ICAO addresses encoded according. The Control Byte was included for AVLC frame completeness but was not used for the RIPS tests.

Table 4 CPDLC Message Type Definition

Value	Message Type Definition
0x11	CPDLC Uplink Message
0x12	C-CAST Status
0x13	C-CAST Downlink Status (CPDLC Downlink Acknowledge)
0x21	CPDLC Downlink Message
0x22	Onyx Status
0x23	Onyx Uplink Status (CPDLC Uplink Acknowledge)

A CPDLC uplink or downlink message contained three other information fields; Message Number, an ASCII Aircraft ID, and the CPDLC data packet described previously. The Message Number uniquely identified the CPDLC uplink or downlink on a short-term basis for acknowledgement purposes. The Aircraft ID field, though filled with data, did not serve a useful purpose for the RIPS tests.

In addition to the main uplink and downlink messages that carry the CPDLC data packets, two other message categories were defined: Status Messages and Acknowledgement Messages. Status Messages were continuously transmitted at 15 Second intervals by both the C-CAST and Onyx mainframe. These transmissions indicated that the ground or airborne components were operational. They also provided a method to log data at constant intervals to analyze the performance of the datalink. In a practical sense, the Onyx status messages enabled C-CAST to track the aircraft in a position-less "CPDLC-based tracking" mode and send taxi instructions when the aircraft was on the West end of DFW (American Airlines hanger) where the RIPS traffic surveillance system did not operate.

Acknowledgement Messages were sent immediately following the receipt of a valid CPDLC packet. The Acknowledgement Message referred to the received CPDLC message's Message Number along with a Received Message Status byte. The Received Message Status byte would be non-zero if the receiver had reported low confidence for the received message, which represented a request to re-transmit that message. C-CAST, upon receipt of an acknowledgement with a Received Message Status value of zero, changed the color of the corresponding ATC instruction on the flight strip to indicate to the controller that the instruction was properly received by the aircraft.

A two-byte Checksum field was added to each CPDLC message before being sent out to any node for data integrity checking purposes. For all messages received by a VDL Mode 2 receiver, an Integrity Byte was attached to the end of the CPDLC message packet. The Integrity Byte was a summary of the VDL Mode 2 received burst quality reporting parameters generated by the receiver. The datalink manager software set the corresponding bits of the Integrity Byte when the parameters exceeded a preset threshold. The Integrity Byte definition is given in Table 5.

Table 5 Integrity Byte Definition

Bit #	Message Type Definition
7 (msb)	Broken message
6	FEC algorithm failure
5	Confidence factor threshold not met
4	Not used (0)
3	Not used (0)
2	Not used (0)
1	Signal quality threshold not met
0 (lsb)	Signal strength threshold not met

6.4   Ground Station

The ground station for the CPDLC datalink was located within the East Tower compound. Figure 5 CPDLC Ground Station Equipmentshows the radio equipment and host computer that was located inside the temporary RIPS trailer.
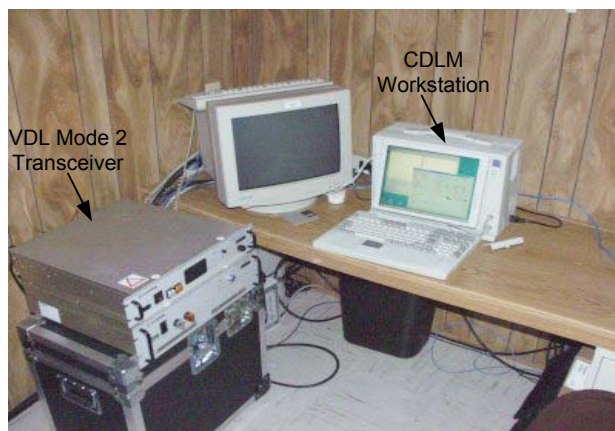

Figure 5 CPDLC Ground Station Equipment

Figure 6 CPDLC Ground Station Antenna

A standard, vertically polarized, VHF aeronautical communications antenna tuned to the assigned frequency was used for the ground station. The antenna was erected along the West side wire fence of the East Tower compound. Antenna height was approximately 16 feet above ground level. Figure 6 CPDLC Ground Station Antenna shows the CPDLC ground station antenna.

6.5   Airborne Unit

The airborne radio equipment was identical to that of the ground station. The radios were installed in the Technology Transfer Area 2 (TTA2) equipment palette situated near the aft of NASA's Boeing 757 Airborne Research and Integrated Experimental System (ARIES) aircraft. A 300 MHz Pentium II™ based FieldWorks™ workstation running the Windows-NT™ operating system hosted the Airborne Datalink Manager (ADLM). Figure 7 shows TTA2 and the ADLM host computer. The airborne antenna was a standard blade-type wide-band VHF aeronautical communications antenna. Figure 8 shows the antenna's location on ARIES. The ADLM connected to the Onyx mainframe via standard Ethernet cabling, which in turn routes to a fiber-optic I/O ring known as the ScramNet. For time stamping purposes, the ADLM workstation was time synchronized using the aircraft's IRIG-B network.
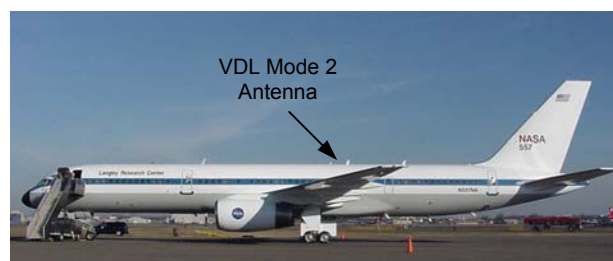

Figure 7 TTA2 Showing ADLM Workstation

Figure 8 ARIES Showing VDL Mode 2 Antenna

6.6   CDLM/ADLM Design Details

The CDLM and ADLM software performed the interfacing between the Harris VDL Mode 2 radios and the C-CAST and Onyx mainframe respectively. Originally the CDLM and ADLM were to be developed as two separate entities. However since their functionality was similar, the software was written such that it could be configured to behave as either the CDLM or the ADLM. Hence, the following description of the ADLM also applies to the CDLM.

The ADLM was written in C++ using the Borland C++ Builder software development environment. Borland's Rapid Application Development (RAD) technology enabled the ADLM to be developed easily within a limited time frame. Functional blocks within the ADLM were developed as software objects that

facilitated these objects to be reused. One of the main functions of the ADLM was to interface with the Harris radios using their binary RS-232 data I/O ports. The complete RS-232 protocol to interface to the radios and all handshaking and error detection functions were integrated into a single software object. This object was written so that it could be configured to behave as the interface for either the receiver or the transmitter. Hence the ADLM used two instantiations of this object to interface to the combined transceiver. The ADLM used the TCP/IP client socket objects available in the Borland development suite to implement the Ethernet client socket. When a message was received from the Onyx, the ADLM first performed a checksum validation to determine that the message was received correctly. It then parsed the Message Type byte to determine what operation was to be performed with the message. If the message was meant for transmission, the ADLM re-arranged the message into the format required by the transmitter and invoked the transmitter's Fill Buffer command, which sends the data packet to the transmitter. Once a message is transmitted, the transmitter sends an acknowledgement that the message (with a given reference number) was transmitted successfully. The ADLM checked for this acknowledgement to verify that the message was successfully sent.

When a message arrived at the receiver, it sent the received AVLC packet to the ADLM. The ADLM verified that the message was valid through its checksum. The ADLM also performed various ambiguity resolution routines to extract the AVLC frame. The AVLC frame was then packaged into the CPDLC message format given in Table 1. Upon receiving a message, the receiver outputs various VDL Mode 2 parameters pertaining to the quality of the received RF burst. The ADLM compared these parameters to preset thresholds and attached the resulting Integrity Byte (Table 5) to the end of the CPDLC message before it was sent to the Onyx.

In addition to the basic error checking, re-packaging, and relaying of CPDLC messages between the Onyx and the VDL Mode 2 transceiver, the ADLM performed configuration functions such as setting frequency, output power, VDL Mode 2 link parameters, etc. for the radios. The ADLM also time stamped and logged all received and transmitted messages, all VDL Mode 2 parameters reported by the receiver, and any other messages reported by the radios for data analysis.

For datalink debugging purposes the ADLM incorporated several test modes to independently check each applicable part of the CPDLC datalink. These included automatic generation and transmission of CPDLC "ROGER" downlink messages at a configurable interval, automatic acknowledgement of a CPDLC uplink message by transmitting the corresponding downlink "ROGER" message, and modes to perform the equivalent functions to test the TCP/IP link.

The ADLM's Graphical User Interface (GUI) was divided into four main panels, as shown in Figure 9. These allowed a user to see at a glance, the operational mode of the ADLM and the state of the receiver, transmitter, and TCP/IP link in real time. Activity indicators displayed RS-232 and TCP/IP traffic between the transmitter, receiver, and Ethernet link by changing color (between yellow and green) whenever messages were sent or received by these components. The tab sheet at the bottom of the GUI allowed the user to monitor the CPDLC messages flowing through the ADLM in either binary, transcript, decoded CPDLC, or log file formats. These features helped to debug numerous problems during the many RIPS system checkout phases and enabled the user to verify that the CPDLC datalink was operating properly during all flight tests.
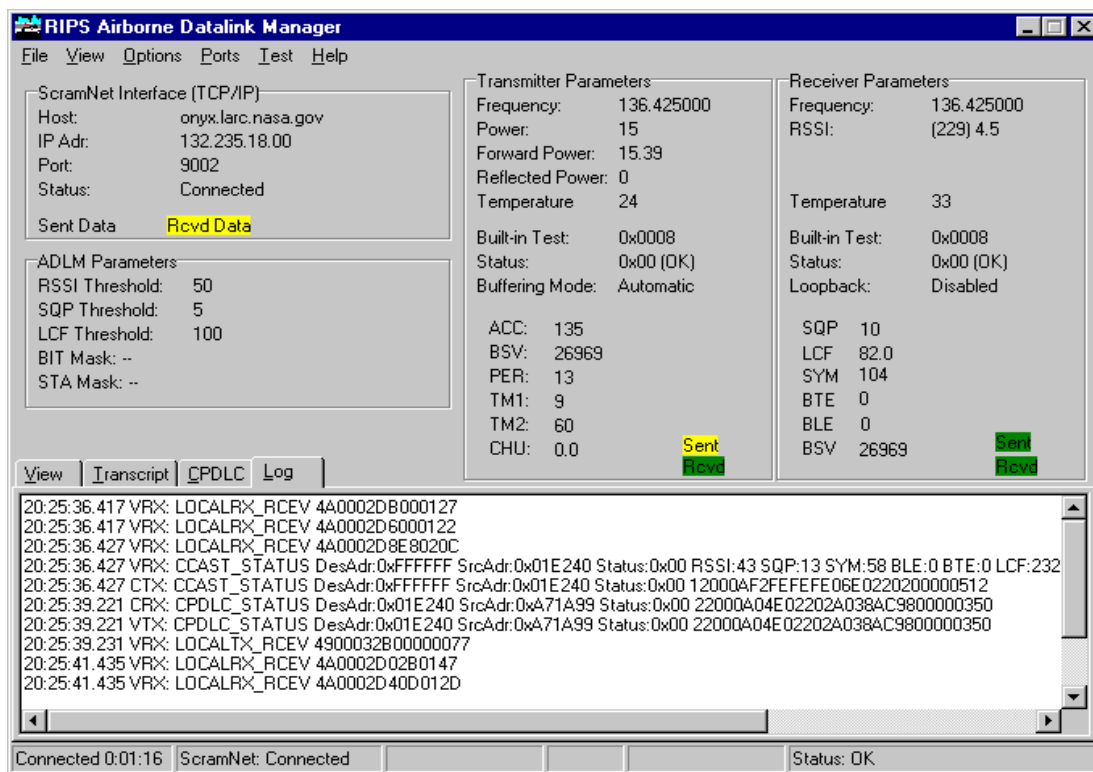
Figure 9 CDLM/ADLM User Interface

7   Voice Recognition

Originally C-CAST relied on a Verbex voice recognition card. Initially obtained in 1996, the cards proved to be extremely reliable; in the 1997 Atlanta tests, the Verbex system had a 97 percent accuracy rate. While dependable, the system had limitations: Verbex only supported DOS drivers and lacked support for the Win32 API platform. The next step was to implement dynamic call signing and complex, multi-instruction messages; it became clear that the speech recognition system needed to be updated. The Verbex system also required users to train for up to one hour before using the program; while this ensured higher accuracy rates; the training period was inconvenient and required considerable time and effort in maintaining user's voice profiles.

In the fall of 1999, after careful consideration, the Lernout & Hauspie (L&H) speech recognition engine was chosen to replace the Verbex system. The L&H engine worked well with Windows-based applications and supported the large, complex grammar files that were needed. The grammar file could be easily modified to allow for multiple pronunciations of a single word or phrase, which eliminated users' training time and allowed users with discrete accents to readily use the application. The software could easily handle the demands of dynamic call signing.

7.1   Voice recognition issues
As a relatively new technology, voice recognition still poses several challenges. While time and development may minimize or even eliminate these issues, currently they remain potential obstacles to successful implementation of voice recognition. By giving careful consideration to these issues, these "obstacles" can be overcome.

Voice recognition systems often utilize grammar files; a grammar file defines the syntax of what will be said in the use of this system. Also included are any special pronunciations, which will be discussed later. From this grammar file the voice recognition system is able to compile lexical trees (Figure 10 and

Figure 11), which the system uses to recognize a statement, by parsing the tree and matching words to the syntax defined by the tree and grammar file.
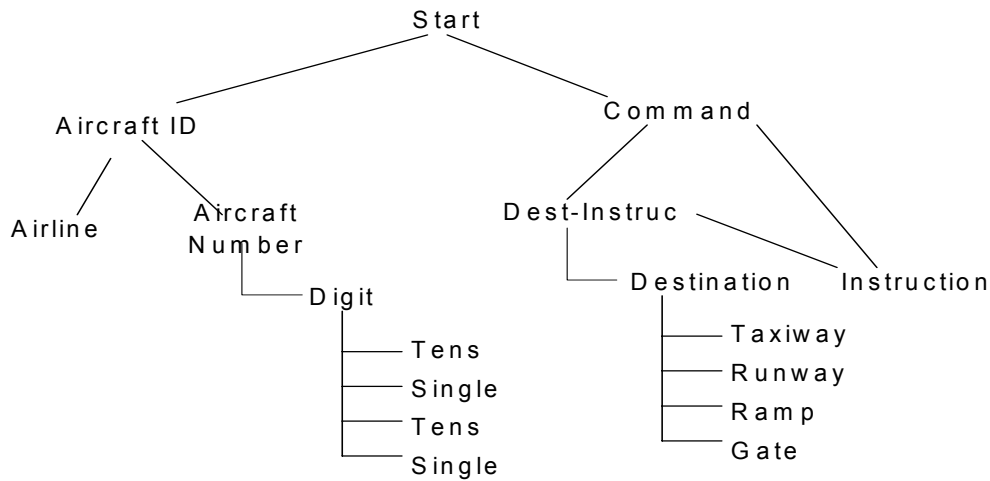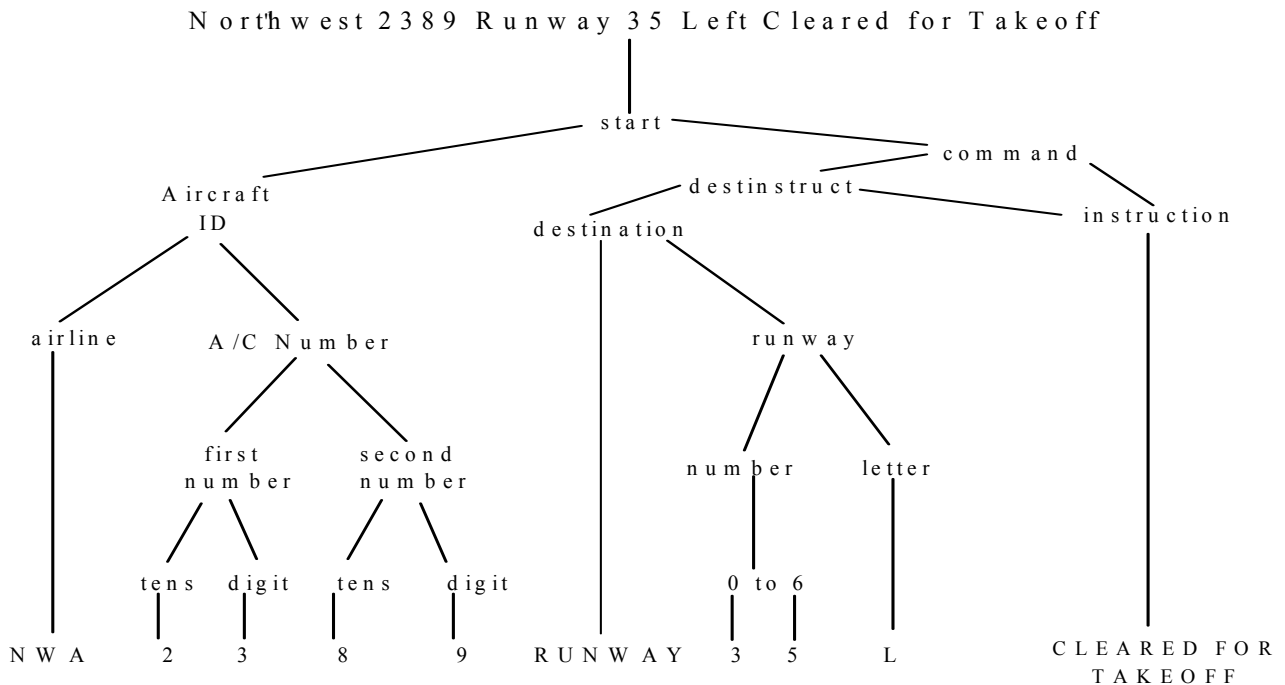
Figure 10 Generic Lexical Tree

Figure 11 Completed Lexical Tree

Voice recognition engines are capable of accepting and correctly interpreting most subtle differences in accent. However, for extreme variations in pronunciation, alternatives must be defined within the grammar file. For example, controllers use the term "Kaybec" rather than "Quebec" as the phonetic name for the letter "Q"; the system should return "Quebec" as the result of both pronunciations. "Kaybec" should be designated as an alternative pronunciation for "Quebec". Multiple dialects and accents are a factor in deploying an application incorporating speech recognition.

Multiple accents can also reduce accuracy; extreme variations in word pronunciation can cause a sensitive speech recognition engine to reject correct input. While individual words can be altered to accept multiple versions, redefining many words in large vocabularies might be too time consuming. In this case, the engine accuracy rate can be reduced to accept a larger of variety of input, in situations such as ATC communications, care must be taken to ensure the accuracy of the instructions.

7.2   Complex and varied grammars/terminology

One of the more difficult aspects of implementing voice recognition is the creation, refinement and maintenance of the grammar file. The file must contain all possible phrases and commands that might be uttered to the system. The terminology and layout of all possible phrases must be rigidly defined and strictly adhered to by users that include everything from the simplest acknowledgement to the most complex taxi route command. Table 1and Table 2 show sample messages that were incorporated into the system design.

Several guidelines were used to design the system grammar. First, the grammar file should be as simple as possible. Longer phrases can be broken up into smaller, more manageable chunks. The use of "wake-up words" can help avoid errors and mangled outputs, especially in complex grammars. If one word or phrase type always precedes another, the speech recognition engine can use the former as a marker, improving recognition times. For example, when directing an aircraft to particular part of the airport, we know that the controller will tell the aircraft "CROSS" followed either by a Taxiway or Runway. Therefore, we can write the following in the grammar file:

<COMMAND>:
: CROSS RUNWAY <runwaynumber>
: CROSS TAXIWAY <taxiwayname>;

as opposed to:

<COMMAND>
: CROSS <location>;

In the example, <runwaynumber>, <taxiway name>, and <location> represent a multitude of further options available to the user. Runway numbers and taxiway names are formatted differently, allowing the speech recognition engine to differentiate between the two. Instead of trying to choose between every runway and taxiway available, the number of possible choices has been narrowed to two, reducing the chances of an erroneous message. In essence, we break one large statement into a multitude of smaller statements.

Grammar files should only incorporate meaningful results; if a specific range of values is used, the grammar file can exclude choices outside of the accepted range. For example, frequencies for control towers should not consider negative values. Making a grammar file accept generic values might be tempting, especially with large files, but will reduce accuracy rates and slow recognition times. Therefore, we should ensure, whenever possible, that only meaningful results are allowed for our grammar file. Occasionally, situations will arise when a generic portion of the grammar file is needed. One example is airplane's call sign. A list of all possible call signs would not only be extremely large (potentially wasting space) but would also present the voice recognition engine with a extreme range of options (many of which sound very similar), increasing the likelihood of inaccurate recognition.

The list of possible aircraft numbers (IDs) is broken down into several manageable pieces by using dynamic call signing,. The first part of the call sign is the airline itself. Each abbreviated airline name is matched with the user's input; if "Northwest" was spoken, the speech recognition engine would return "NWA". The ID number is handled using known constraints. Since an airline call sign number has a maximum of known digits and each digit ranges from 0 to 9, the engine is instructed to listen for up to four

digits, then truncate the four separate digits into one and add to the aircraft name to create the complete call sign. Figure 6 shows an example of user input and the corresponding output.

> **_Dynamic Call Signs_**
>
> 'NASA five five seven"
>
> is interpreted as:
>
> "NASA557"

This method of simplifying potentially crippling complexity can be applied in many different situations. While it is a simple solution to a complex problem, it is not a universal one. In some cases, simply listing the possibilities will be the better option. Careful consideration should be given for when a generic solution (such as call signing) is appropriate.

7.3   Accuracy

Maintaining accuracy rates is a significant challenge in developing speech recognition applications. Two key issues exist when considering accuracy rates; first, the precision rates must be within acceptable levels for the particular system. Additionally error-handling measures are necessary as these systems are not perfect and some errors will occur. Error levels vary with each system's purpose. Voice dictation programs for word processors allow users to correct their mistakes; a lower degree of accuracy is needed with this software. Safety critical systems, such as ATC communications and others involving sensitive data require the highest degree of accuracy possible which should be decided upon during the system's design phase.

7.4   Hardware restrictions

Speech recognition systems place significant demand on processor and memory resources; system usage is directly proportional to the size and complexity of the grammar file. Resource allocation is a significant concern when designing applications with speech recognition. C-CAST showed a marked performance decrease when tested on systems with Pentium II 300 MhZ or less as the speech recognition features interfered with the sending and receiving of messages across the TCP/IP connection. Computers using speech recognition programs must be equipped with a sound card compatible with the system and an intake device, such as a microphone.

7.5   User-friendly vs. Functionality

Dynamic call signing, properly implemented, is a good example of a balance between functionality and user-friendliness. By allowing the system to determine at that moment the aircraft's call sign we are accomplishing both goals. However, care must be taken so that the system is robust enough that it maintains a zero or very low rate of errors.

The FAA's standard format for controller instructions to aircraft determines that the most common phrases begin with the aircraft ID followed by a series of instructions. Even though a single message may contain many instructions, using compound messaging allows for a large variety of messages with a high accuracy rate and minimum demands on processor time.

Compound messaging uses many of the techniques employed in dynamic call signing. In a message, the aircraft ID is followed by one or more instructions. If each command to the aircraft is represented by <instruction> in the grammar file, the top level of the grammar file would contain:

> <aircraftid> !repeat(<instruction>)

The aircraft is identified first, then the remainder of the message is broken down into a series of instructions. Instead of trying to identify a single large message, the engine will repeatedly decipher each instruction. At project runtime this structure is only slightly more complex than processing a single instruction, with minimal overhead in processor time involved. As has been shown, a system can be made to be user-friendly while still preserving its functionality. With the proper planning and creativity, solutions to seemingly critical problems can be found.

7.6   User-Training

Early speech recognition systems required the user to spend significant time working with the program to achieve higher accuracy rates. By creating their own profile, users enjoyed a high level of accuracy; however, because the sample was taken over a single training session, the quality and timbre of the user's voice was so narrow that very little variation was allowed. If the individual had a cold or other condition that changed their voice, the user profile would be useless, requiring further training or tinkering with the system to increase accuracy.

Recent advances in technology have allowed speech-recognition engines to become more powerful and sensitive, returning correct values in a multitude of environments. User-training is now optional and only necessary under two conditions: if a word or phrase is consistently misinterpreted by the engine, or if the user's voice or accent is so remarkable that a unique file needs to be created. If users are having difficulty, they simply might require more time; as users became familiar with C-CAST, their accuracy rates and comfort with using the system improved significantly.

8   Testing

C-CAST has undergone several testing procedures. The first of a series of tests occurred at NASA-Langley Research Center in Virginia in August of 2000. These tests were conducted to ensure proper installation of equipment on the test aircraft, radio communication verification and encoding and decoding of manual text messages. A week later, the aircraft radios were installed in a test van to simulate an aircraft taxiing on the runways. This test was conducted at Dallas-Fort Worth Airport and included radio coverage, system integration and error detection.

A few months later, in October 2000, flight tests were conducted at Dallas-Fort Worth Airport. These tests were conducted to determine the communication limits of the radios and system integration. Later in the month, public demonstrations were conducted for NASA, FAA and other industry professionals. The results and configurations are described in the following sections.

8.1   Langley August Tests

The test configuration included a Boeing 757 aircraft with the airborne equipment installed on the aircraft. At first, the radios were unable to communicate and it was later determined that the DLM must be used on an NT platform due to some specific API calls that are not defined in Windows 98. After this problem was determined and fixed, communication between the radios went smoothly. A test of the encoding and decoding process was initiated. Sending all possible messages using the manual interface did this. A few bugs were found and the differences were worked out before the next round of testing.

Overall the tests were very successful. Lessons were learned about system debugging and integration. This round of tests also saved a considerable amount of money by verifying the message encoding process. It was more economical to work the problems out during the ground tests rather than during the much more expensive flight tests held in October 2000.

8.2   Dallas August Tests

These tests were primarily meant to test the radio coverage abilities of the VDL Mode-2 radios. They also served the purpose of making sure that all systems were in place and communicating with each other properly. C-CAST was connected to the Network Time Protocol, the Surveillance Server and Ohio University's DLM. The Surveillance Server was connected to Trios' Data Link Manager.

The test vehicles were driven along the runways at high speeds (up to 80 mph) over the entire airport. The ground stations were located in a trailer by the East tower. A view of the radio coverage is located in Figure 3 and Figure 4. It can be seen from the figure that the strongest coverage was near the East tower, all the while still receiving good coverage everywhere on the airport surface. Adequate coverage was obtained over the entire airport surface.

8.3   Dallas October Tests

The October tests in Dallas were the actual flight tests and were meant to test the C-CAST system to its limits. The airborne equipment was installed in the Boeing 757 aircraft and the aircraft was flown for data collection. This consisted of 5 nights of data collection and 5 nights of checkout. The radios proved to be adequate for the needs of the C-CAST system, allowing communication to be established well before the aircraft reached approach distance. All communication went well and messages were transmitted and decoded successfully.
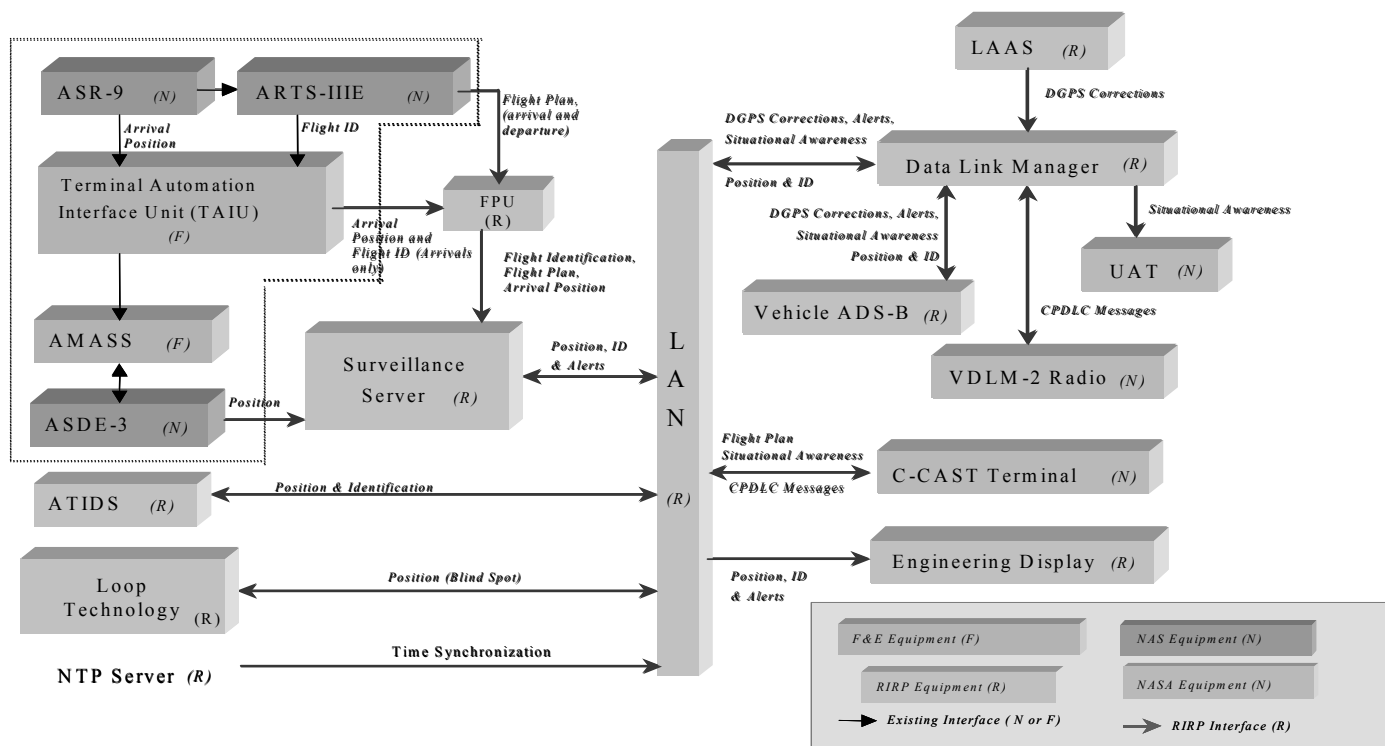

Figure 12   DFW Ground Infrastructure

9   Future Work

There is strong potential for future work in the area of Controller/Pilot communications technology. Among these developments is the use of a "Heads-up" virtual display for pilots and controllers. This allows the users to see messages or even routes mapped out for them, superimposed on their current view, there-by creating a truly "heads-up" environment for the user.

Another future development could be the mapping of the spoken command on a digital map for the controller so that the route can be verified before transmission. This might be a little easier to review rather than the spoken message.

Some day a user might be able to simply draw the route on a digital map and then hit a button to encode the message into text, synthesized voice and even a copy of the digital route displayed on a map. This would drastically shorten the time it takes to encode and send a message and might be less prone to errors since a visual reference would be provided.

10 Summary

The tests that were conducted from August to October of 2000 were very successful for the RIPS researchers. The system proved to be reliable under the testing conditions and showed the types of potential systems that could be made available to the ATC community. The types of technologies that were developed under these NASA projects have the potential to act as stepping-stones to the future of ATC technology.

Consistent with the conclusions drawn from the coverage tests, the datalink worked flawlessly during all five data collection, and two system demonstration flights. During all flight tests, both the CDLM and ADLM consoles were monitored for proper operation of the radios, the software, and the datalink in general. No critical failure of either the radios or the software was detected. All CPDLC uplink, downlink, and acknowledgement messages were sent and received correctly as best monitored through the consoles. Attempts are underway to analyze all data collected during the flight tests. Preliminary results show that a total of 1021 CPDLC uplink and downlink messages were transmitted for all five data collection flights. Of these, the analysis shows a total of 5 messages failed to be received by either the C-CAST or the Onyx. Hence the datalink had an overall reliability of 99.51%. Conclusive analysis is underway to determine which section of the datalink was responsible for these dropped messages. Overall, the VDL Mode 2 datalink described in this paper performed above expectations to support CPDLC operations for all RIPS flight tests performed at DFW.

Several advantages are readily apparent when considering implementing speech-recognition features in ATC applications. One consideration is that speech recognition employs a "hands-off" approach. Users speak control instructions into a headset equipped with a microphone. Because controllers are familiar with this setup, a minimum amount of time would be needed to familiarize users with the speech-recognition system. The combination of a speech-recognition engine and a graphical user interface (GUI) interface would be more efficient; the interface would provide ATCs with the plane's position on the runway, allowing for immediate corrections, if needed. Finally, speech-recognition has advanced to the point that variations in pronunciation and inflection are easily accommodated, allowing the software to be used on a broad scale in many regions.

The voice recognition system performed well even though the ambient noise in the room sometimes was loud. The grammar file was generic with minimal tailoring to the specifics of the Dallas-Ft. Worth airport to enable the team to test dynamic phrasing. The C-CAST was successful at providing the NASA 757 with taxi route information via Controller-Pilot Datalink Communications. The requirement to use standard phraseology proved to be the main shortcoming with the voice recognition system as implemented. Air traffic controllers as well as other visitors during the tests and demonstrations thought that this type of system could enhance surface operations at a variety of airports.

11 References

[1]   Best, Eric, and Rankin, James M. *Controller-Pilot Data Link Results from NASA and FAA Dallas-Fort Worth 2000 Test and Demonstration*, 20th Digital Avionics Systems Conference, Daytona Beach FL, October 2001.
[2]   Aarons, Richard N., "Controller/Pilot Data Link", *Business & Commercial Aviation*, February 2001. pp. 44 – 7.
[3]   Gunawardena, Sanjeev, and Rankin, James M. "Controller – Communication and Situational Awareness Terminal (C-CAST) to Data Link Manager (DLM) Interface Control Document" Revision 2.0. July 27, 2000.

[4]   Gunawardena, Sanjeev, *Controller – Pilot Communications Using a VDL Mode 2 Datalink for the NASA Runway Incursion Prevention System* , Proceedings of the 20th Digital Avionics Systems Conference, Daytona Beach FL, October 2001.

[5]   Lechner-Stoeckel, Alicia; Ecker, Kevin; and Mattson, Pat. *Voice recognition – Software Solutions in Real Time ATC Workstations*. 20th Digital Avionics Systems Conference, Daytona Beach FL, October 2001.

[6]   *Minimum Operational Performance Standards for ATC Two-Way Data Link Communications*, RTCA DO-219.

[7]   *RIRP/DFW-NASA Demonstration*, Trios Associates, Inc., May 2, 2000.

[8]   RTCA, *DO-224A Signal-in-Space Minimum Aviation System Performance Standards (MASPS) for Advanced VHF Data Communications Including Compatibility with Digital Voice Techniques*.

[9]   International Civil Aviation Organization (ICAO), 1999, *Manual of Technical Provisions for the Aeronautical Telecommunications Network (ATN)*, 2nd edition.

[10]   Rankin, James, Jul. 2000, *C-CAST Message Set for DFW RIPS Demonstration*, Avionics Engineering Center, Ohio University.

[11]   Gunawardena, Sanjeev, and Rankin, James. *Airborne Datalink Manager to NASA-Langley B757 ScramNet Interface Control Document*, Avionics Engineering Center, Ohio University. Jul. 2000.

[12]   Young, Steve, and Rankin, J., *Improving the Safety of Airport Surface Operations*, Air Traffic Technology International, Nov. 1998.

[13]   Rankin, James, and Mattson, P. *Controller-Pilot Data Link Statistics from NASA's 1997 Atlanta Flight Test*, Proceedings of the 17th DASC, 1998.

[14]   Rankin, James, and Mattson, P. *Controller Interface for Controller-Pilot Data Link Communications*, Proceedings of the 16th DASC, 1997

12 Acronyms

AEC	Avionics Engineering Center at Ohio University
AVLC	Aviation VHF Link Control
C-CAST	Controllers' Communication and Situational Awareness Terminal
CSMA	Carrier-Sense Multiple Access
CI	Controller Interface (system used in Atlanta in 1997)
CPDLC	Controller-Pilot Data Link Communications
DFW	Dallas-Fort Worth International Airport
H-SALT	Hold Short and Landing Technology
L&H	Lernout & Hauspie Voice Recognition System
OU	Ohio University
RIPS	Runway Incursion Prevention System.
SCSU	St. Cloud State University
SVS	Synthetic Vision System
VDL-Mode 2	VHF Data Link. Mode 2 uses a CSMA protocol. Also, VDLM2